

Лабораторная работа №1, Часть 2: Основы трекинга экспериментов с использованием MLflow

Цель работы: Освоить базовые принципы работы с платформой MLflow для управления жизненным циклом машинного обучения (MLOps). Получить практические навыки логирования параметров, метрик и артефактов вычислительного эксперимента, а также организации их хранения, визуализации и сравнения.

Стек технологий:

- **Операционная система:** Ubuntu 24.04 LTS
 - **Менеджер пакетов и окружений:** Conda (окружение `mlops-lab` из Части 1)
 - **Библиотеки:** `mlflow`, `scikit-learn`, `pandas`, `matplotlib`
 - **Платформа для трекинга:** MLflow Tracking Server (standalone)
-

Теоретическая часть (краткое содержание)

1. Введение в MLOps и MLflow MLOps (Machine Learning Operations) — это совокупность практик, направленных на автоматизацию и надежность жизненного цикла машинного обучения (развертывание, мониторинг, управление данными). Ключевая проблема, которую решает MLOps — обеспечение воспроизводимости, отслеживаемости и управляемости ML-экспериментов. **MLflow** — это open-source платформа для управления end-to-end жизненным циклом машинного обучения. Она включает в себя четыре основных компонента:

- **MLflow Tracking:** API и UI для логирования параметров, метрик, кода и артефактов (графики, модели) в процессе выполнения ML-кода.
- **MLflow Projects:** Стандартный формат упаковки ML-кода для обеспечения воспроизводимости на любой платформе.
- **MLflow Models:** Стандартный формат упаковки ML-моделей для упрощения их развертывания с помощью различных инструментов.
- **MLflow Model Registry:** Централизованное хранилище моделей для управления их жизненным циклом (staging, production, archiving).

В данной работе focuses на компоненте **Tracking**.

2. Ключевые концепции MLflow Tracking

- **Эксперимент (Experiment):** Контейнер для группы запусков (например, "Оптимизация гиперпараметров для модели X").
- **Запуск (Run):** Одно выполнение кода, которое логируется в MLflow. Каждый запуск фиксирует:
 - **Параметры (Parameters):** Входные переменные модели (например, `max_depth`, `learning_rate`).
 - **Метрики (Metrics):** Количественные показатели качества модели (например, `accuracy`, `rmse`). Метрики могут обновляться по ходу выполнения (эпохам, итерациям).
 - **Артефакты (Artifacts):** Файлы любого типа, связанные с запуском (например, графики, обученная модель, файл с предсказаниями).

- **Теги (Tags):** Произвольные ключ-значения для пометки запусков.
 - **Backend Store:** Хранилище (файловая система или база данных), где сохраняются метаданные запусков (параметры, метрики).
 - **Artifact Store:** Хранилище (например, локальная папка, S3) для артефактов.
-

Задание на практическую реализацию

Этап 1: Установка MLflow и запуск Tracking Server

1. Активация окружения и установка пакетов:

- Активируйте созданное ранее окружение `mlops-lab`:

```
conda activate mlops-lab
```

- Установите библиотеку `mlflow`:

```
pip install mlflow
```

2. Запуск MLflow Tracking Server:

- В терминале выполните команду для запуска сервера. Флаги `--backend-store-uri` и `--default-artifact-root` указывают, где хранить метаданные и артефакты соответственно (в данном случае — в локальной директории `./mlruns`).

```
mlflow server --backend-store-uri sqlite:///mlflow.db --default-artifact-root ./mlruns --host 0.0.0.0 --port 5000
```

- Сервер будет запущен и доступен в браузере по адресу `http://<your_ip_address>:5000` или `http://localhost:5000`. Вы должны увидеть интерфейс MLflow UI с списком экспериментов (пока пустой).

Этап 2: Написание и запуск скрипта с трекингом эксперимента

1. Создание Python-скрипта:

- Создайте файл `mlflow_basic.py`.
- Скопируйте в него следующий код, который реализует простое обучение модели логистической регрессии на встроенном в `sklearn` наборе данных Iris.

```
import mlflow
import mlflow.sklearn
from sklearn.datasets import load_iris
```

```

from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_score,
f1_score
import matplotlib.pyplot as plt

# Установите URI для отслеживания (указывает на запущенный сервер)
mlflow.set_tracking_uri("http://localhost:5000")

# Создайте или установите активный эксперимент
experiment_name = "Iris_Classification_Baseline"
mlflow.set_experiment(experiment_name)

# Загрузка данных
iris = load_iris()
X = iris.data
y = iris.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Определите параметры модели для логирования
params = {
    "solver": "lbfgs",
    "max_iter": 1000,
    "multi_class": "auto",
    "random_state": 42
}

# Начало запуска MLflow
with mlflow.start_run():
    # Логирование параметров
    mlflow.log_params(params)

    # Создание и обучение модели
    model = LogisticRegression(**params)
    model.fit(X_train, y_train)

    # Предсказание и расчет метрик
    y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred, average='weighted')
    recall = recall_score(y_test, y_pred, average='weighted')
    f1 = f1_score(y_test, y_pred, average='weighted')

    # Логирование метрик
    mlflow.log_metric("accuracy", accuracy)
    mlflow.log_metric("precision", precision)
    mlflow.log_metric("recall", recall)
    mlflow.log_metric("f1_score", f1)

    # Логирование модели
    mlflow.sklearn.log_model(model, "model")

    # Создание и логирование артефакта (графика)

```

```

fig, ax = plt.subplots()
ax.bar(['Accuracy', 'Precision', 'Recall', 'F1'], [accuracy, precision,
recall, f1])
ax.set_ylabel('Score')
ax.set_title('Model Performance Metrics')
plt.savefig("metrics_plot.png") # Сохраняем график в файл
mlflow.log_artifact("metrics_plot.png") # Логируем файл как артефакт

# Вывод метрик в консоль для удобства
print(f"Accuracy: {accuracy:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
print(f"F1 Score: {f1:.4f}")

print("Run completed and logged to MLflow!")

```

2. Запуск скрипта:

- Убедитесь, что MLflow Server запущен.
- В другом окне терминала (с активированным окружением `mlops-lab`) выполните:

```
python mlflow_basic.py
```

- В консоли должны отобразиться рассчитанные метрики и сообщение об успешном завершении.

Этап 3: Анализ результатов в MLflow UI

1. **Откройте UI:** Перейдите по адресу <http://localhost:5000>.
2. **Найдите свой эксперимент:** В боковом меню выберите эксперимент

[Iris_Classification_Baseline](#). Вы увидите список запусков (в данном случае один).

3. Изучите детали запуска:

- Нажмите на дату/время запуска, чтобы открыть его детальную страницу.
- Во вкладке **Overview** просмотрите логированные параметры и метрики.
- Перейдите во вкладку **Artifacts**. Здесь вы должны увидеть:
 - Папку `model` с сохраненной моделью в формате MLflow.
 - Файл `metrics_plot.png` с построенным графиком.

4. Сравнение запусков (опционально, для будущих работ):

- Запустите скрипт еще 1-2 раза, изменив какой-либо параметр (например, `"solver": "liblinear"`).
- В UI на странице эксперимента можно будет сравнить метрики всех запусков.

Требования к оформлению и отчету (для Части 2)

Критерии оценки для Части 2:

- **Удовлетворительно:** Успешно выполнен Этап 1 (установка MLflow, запуск Tracking Server) и Этап 2 (написание и выполнение скрипта `mlflow_basic.py` без ошибок). В UI MLflow отображается один завершенный запуск.
 - **Хорошо:** Дополнительно к критерию "Удовлетворительно" в UI MLflow корректно отображаются все логированные параметры, метрики и артефакты (модель и график). Продемонстрирована работа UI.
 - **Отлично:** Все задания выполнены в полном объеме. Проведен дополнительный эксперимент: скрипт был модифицирован для логирования другой метрики (например, `log_loss`) или параметра (например, `test_size`), произведено 2 и более запусков, и в UI продемонстрировано их сравнение.
-

Рекомендуемая литература (источники)

1. **Официальная документация MLflow:** <https://mlflow.org/docs/latest/index.html>
2. **Официальный туториал MLflow Tracking:** <https://mlflow.org/docs/latest/tracking.html>
3. **Scikit-learn User Guide:** https://scikit-learn.org/stable/user_guide.html
4. **Статья "Getting Started with MLflow":** <https://towardsdatascience.com/getting-started-with-mlflow-52eff9c57d0d>
5. **Книга "Machine Learning Engineering" by Andriy Burkov:** (Главы, касающиеся MLOps и воспроизводимости экспериментов)
6. **MLflow: A Tool for Managing the Machine Learning Lifecycle:**
<https://databricks.com/blog/2018/06/05/introducing-mlflow-an-open-source-machine-learning-platform.html>
7. **Документация по sklearn.metrics:** <https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics>
8. **Matplotlib Documentation:** <https://matplotlib.org/stable/contents.html>